International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Feb-2021)
http://dx.doi.org/10.12785/ijcds/100137

# Prioritizing CWE/SANS and OWASP Vulnerabilities: A Network-Based Model

**Basim Mahmood**[1]

[1]*Department of Computer Science, University of Mosul, Mosul, Iraq*
[1]*BioComplex Laboratory, Exeter, UK*

*Received 25 Jul. 2020, Revised 8 Aug. 2020, Accepted 25 Dec. 2020, Published 8 Feb. 2021*

**Abstract:** Nowadays, software applications have become ubiquitous and a centric need in our life. Most of our business, education, and social activities cannot be performed without software applications. Moreover, the development of software has become the main focus in the market due to the wide variety of customer needs. However, the vast amounts of software that are distributed around the world have dangerous weaknesses and vulnerabilities that can be exploited by cybercriminals to get unauthorized access to users' data. Thousands of cybercrimes are reported every day around the world due to these vulnerabilities. Therefore, it is critically needed to understand software vulnerabilities and the relations among them aiming at having convenient practices against the dangerous attacks and mitigate their impact. This article analyses the weaknesses that have been defined by the CWE/SANS and OWASP, which are considered as the most trusted and accredited cyber-security organizations. These organizations use a specific scoring system called Common Weakness Scoring System (CWSS) for ranking vulnerabilities based on their frequency of broken and other factors. We involve the concepts of complex networks in the methodology of our analysis. To this end, we generate networks each of which represents the CWE/SANS and OWASP top vulnerabilities issued in a particular year. We, then, analyze the generated networks based on network level and node level measurements. The findings show that CWSS can include centrality measurements for ranking vulnerabilities in a more accurate way. Finally, we believe that centrality measurements can play a significant role and can be considered as a powerful tool in improving CWSS in terms of accuracy.

**Keywords:** Data Analysis, CWE/ SANS and OWASP Vulnerabilities, Complex Networks, Software Security.

## 1. INTRODUCTION

The recent years have witnessed a great revolution in the field of software development. Software applications have become an important need for most of our daily and work activities. The number of software developers has significantly increased due to the wide variety of customer needs. However, the vast amount of software distributed around the world has dangerous security weaknesses and vulnerabilities such as Injection and Stack Overflow, which can be exploited by cyber-criminals to get illegal or unauthorized access to users' data [1],[2]. Therefore, the complexity of software design is also increased since thousands of cybercrimes are reported every day around the world [2]. Some security organizations around the world try to help developers with recommendations when designing their software applications. The most popular and trusted organizations that provide unbiased security and practical information on applications are Common Weaknesses Enumeration (CWE) and SANS institute for security [3] (CWE/SANS), and Open Web Application Security Project (OWASP) [4]. These important sources of information have played a crucial role in mitigating dangerous attacks around the world and enriched developers with best security practices. CWE is supported by the National Cyber Security Division of the US Department of Homeland Security. Furthermore, CWE uses codes to denote a particular vulnerability. For instance, they use the term CWE-(vulnerability ID) such as CWE-287 to refer to Improper Authentication vulnerability. CWE also uses a scoring system for assigning a score to every single vulnerability in order to rank vulnerabilities according to their scores (highest ranks means more vulnerable). All the weaknesses under CWE are subject to Common Weakness Scoring System (CWSS). This system depends on the efforts of organizations, developers, and security communities in its mechanism for prioritizing vulnerabilities and assigns a score to each. CWSS is based on metric groups as follows [5]:

i. **Base Finding Metric Group:** includes technical impact, acquired privilege, acquired privilege layer, internal control effectiveness, and finding confidence.
ii. **Attack Surface Metric Group:** includes required privilege, required privilege layer, access vector, authentication strength, level of

*E-mail: bmahmood@uomosul.edu.iq, bmahmood@biocomplexlab.org*

interaction, and deployment scope.

iii. **Environmental Metric Group:** includes business impact, the likelihood of discovery, and the likelihood of exploitation, external control effectiveness, and prevalence.

The collective values of the factors of each group generate the value of the whole group. The multiplication of the values of the three mentioned groups produces the final score of vulnerability. The scores of CWSS take the range of 1 and 100 [5]. The impact of each vulnerability varies from accessing users' data to causing damages to users' files or devices [6]. CWE/SANS and OWASP provide lists of top vulnerabilities periodically, and there are about 20 industry experts who contribute in assessing vulnerabilities and update the list of top vulnerabilities. This paper tries to involve network measurements in the analysis of the relations among vulnerabilities. Below, we list the ones we involve in this article and can be classified in two levels [7],[8]; Network Level, and Nodes Level:

**A- Network-level measurements are:**

- *Average Degree (Avg(D))*: the average degree of the total network nodes. In other words, it represents the average number of connections for network nodes [7].
- *Network Diameter ($N_d$)*: the longest path of all the shortest paths in a network [7].
- *Network Density ($D_s$)*: a dense network is a graph ($G$) that has the maximum number of edges ($E$). It reflects the ratio of the number of the actual network edges to the number of potential edges in that network [8]:
- *Average Clustering Coefficient (c)*: clustering coefficient of a node is the proportion of the frequency of connections among node's neighbors to the maximum number of such connections. For the entire network, c represents the average of the clustering coefficient for all network nodes. The *c* of a vulnerability *I* can be formalized as follows [8].

$$c(i) = \frac{2\left|\{l_{ik}: n_j, n_k \in N_i, l_{ik} \in \mathrm{E}\}\right|}{k_i(k_i-1)} \quad (1)$$

where $l_{jk}$ is a vulnerability between the vulnerabilities $n_j$ and $n_k$. While $N_i$ is the total number of vulnerabilities and $k_i$ is the closest vulnerability in the network. The average clustering coefficient of a network (or a network model) is the average of all the c values of the vulnerabilities of that network.

- *Average Path Length (l)*: the average number of links along with the shortest paths for all network pairs [8]:
- *Girvan-Newman Algorithm*: it is used for detecting clusters (communities) in a network with a modularity level that reflects the strength of these detected communities [15].

**B- Node-level measurements are:**

- *Betweenness Centrality ($C_B$)*: for node *i*, it reflects the number of all the shortest paths from all nodes to other network nodes that pass through node *i*. Nodes with higher values of $C_B$ indicate that most network information passes through these nodes and can be formalized as follows [7]:

$$C_B\,(j) = \frac{\sum_{i \neq j \neq k}(\sigma_{ik}(j))}{\sigma_{ik}(j)} \quad (2)$$

where $\sigma_{ik}$ is the shortest paths between the vulnerability *i* and *k*. $\sigma(j)$ is the number of paths that pass through vulnerability *j*.

- *Closeness Centrality ($C_C$)*: reflecting how close a node to other network nodes. For a node, it is the average length of the shortest paths to all network nodes and can be formalized as follows [7]:

$$C_C(j) = \frac{(N-1)}{\sum_j \mathrm{d(ij)}} \quad (3)$$

where *d(ij)* is the distance between vulnerability *i* and *j*.

- *Degree Centrality ($C_D$):* for a node, it is the frequency of connections to other network nodes.

## 2. RELATED WORKS

As mentioned in the previous section, CWSS has weaknesses points that make developers propose and suggest new techniques for scoring vulnerabilities as proven in [9]. The most recent study on the CWE vulnerabilities was performed in [10]. The authors investigated the relations among vulnerabilities using network measurements. Their study focused on three aspects when dealing with security vulnerabilities namely; research concepts, development concepts, and architectural concepts. The issue of scoring systems of CWSS has not given much attention in the literature. A few articles tried to propose techniques for scoring and prioritizing vulnerabilities taking into consideration the weaknesses point in the CWSS scoring system. This system is considered as a static system in which the score of a vulnerability is still associated with that vulnerability until a new list of scoring is issued. In [11], the authors proposed a dynamic approach instead of the current static system. Their system takes into account two temporal factors; vulnerability index and remediation level. The authors believe that these two factors play an important role in determining the severity of particular security

weaknesses and the impact of any security weakness varies over time. Marcel et al. [12] converted weaknesses scores into the form of probabilities and distribute them to attack paths aiming at having an overall metric for the network. This method was able to make predictions on the security issues of a dynamic network. Other authors tried to develop the current scoring system by taking one or all the three metric groups (see Section 1). Pengsu et al. [13] proposed a new technique when dealing with the first metric group (Base Metric). In their technique, they merged scores in three aspects and added probability, effort, and skill attributes. This makes developers weigh the attributes of the Base Metric group according to their needs. A recent study [14] tried to rank security metrics by incorporating the correlations among them. They did not deal with vulnerabilities themselves; instead, they rank security metrics using statistical techniques. However, in this work, we try to use centrality measurements in ranking the top security vulnerabilities issued by CWE/SANS and OWASP. We believe our approach is accurate since it takes into account the relations among vulnerabilities, which is a new dimension that can be adopted and incorporated into the current scoring system for assessing the risk level of vulnerabilities. This dimension is important because vulnerabilities can be a side-effect (or a cause) of other vulnerabilities [14].

## 3. DATASET COLLECTION

We extracted the datasets used in this work from the CWE/SANS and OWASP organizations. The data is available for researchers and developers on the official website of Common Weakness Enumeration CWE. This data is about the most dangerous security vulnerabilities reported by more than 20 industry experts. The extraction process was performed on a crawler program designed especially for this purpose. This program has the ability to crawl the CWE/SANS and OWASP website and retrieve the information needed in our work. This crawler extracts the data in two levels of depth; *Level 1*, extracts information on vulnerabilities from the targeted links directly (lists of top vulnerabilities). *Level 2*, goes through each vulnerability's hyperlink and extracts all the related vulnerability that will further be used in our analysis. The analysis of this work is based on 7 datasets each of which depends on a particular list in CWE/SANS and OWASP for the years of 2004, 2007, 2009, 2010, 2011, 2013, and 2017 separately. Each dataset includes data from the combination of both the aforementioned levels. Below, we present the top classified security weaknesses [3]:

– OWASP top 10 vulnerabilities in 2004 (see Table 1). Each category under this table includes several vulnerabilities (the total number is 108 vulnerabilities).

– OWASP top 10 vulnerabilities in 2007 (see Table 2). Each category under this table includes several vulnerabilities. The total number of vulnerabilities in this list is 28.

– CWE/SANS and Microsoft SDL top 25 vulnerabilities in 2009 (see Tables 3 and 4). Each category in Table 4 includes several vulnerabilities and the total number of them is 26.

– CWE/SANS top 10 vulnerabilities in 2010 (see Table 5). Each category under this table includes several vulnerabilities (the total number of vulnerabilities in this list is 41).

– CWE/SANS top 25 vulnerabilities in 2011 (see Table 6). The total number of vulnerabilities in this list is 41 distributed into 4 categories.

– OWASP top 10 vulnerabilities in 2013 (see Table 7). Each category under this table includes a list of vulnerabilities. The total number of vulnerabilities in this list is 36).

– CWE top 10 vulnerabilities in 2017 (see Table 8). Each category under this list includes several vulnerabilities. The total number of vulnerabilities in this list is 41).

– CWE top 25 vulnerabilities in 2019 (see Table 17). This list is the most recent issued by CWE.

It should be mentioned that each category or vulnerability in the mentioned tables contains one or more vulnerabilities called related vulnerabilities. A related vulnerability is a vulnerability that may cause or lead to another one. For this reason, it is important to take this kind of information into our dataset.

TABLE 1. CWE VIEW: Weaknesses in OWASP top 10 categories (2004).

| Rank | Category | Title |
|---|---|---|
| 1st | A1 | Unvalidated Input |
| 2nd | A2 | Broken Access Control |
| 3rd | A3 | Broken Authentication and Session Man. |
| 4th | A4 | Cross-Site Scripting (XSS) Flaws |
| 5th | A5 | Buffer Overflows |
| 6th | A6 | Injection Flaws |
| 7th | A7 | Improper Error Handling |
| 8th | A8 | Insecure Storage |
| 9th | A9 | Denial of Service |
| 10th | A10 | Insecure Configuration Management |

TABLE 2. CWE VIEW: Weaknesses in OWASP Top 10 Categories (2007).

| Rank | Category | Title |
|---|---|---|
| 1st | A1 | Cross-Site Scripting (XSS) |
| 2nd | A2 | Injection Flaws |
| 3rd | A3 | Malicious File Execution |
| 4th | A4 | Insecure Direct Object Reference |
| 5th | A5 | Cross Site Request Forgery (CSRF) |
| 6th | A6 | Improper Error Handling |
| 7th | A7 | Broken Authent. and Session Management. |
| 8th | A8 | Insecure Cryptographic Storage |
| 9th | A9 | Insecure Communication |
| 10th | A10 | Failure to Restrict URL Access |

TABLE 3. CWE VIEW: CWE/SANS Weaknesses in the 2009 Top 25 Most Dangerous Programming Errors (3 categories).

| Rank | Category |
|---|---|
| - | Insecure Interaction Between Components |
| - | Risky Resource Management |
| - | Porous Defenses |

TABLE 4. Microsoft SDL and the CWE/SANS top 25 vulnerabilities (2009).

| Rank | ID | Title |
|---|---|---|
| 1st | 20 | Improper Input Validation |
| 2nd | 116 | Improper Encoding /Escaping of Output |
| 3rd | 89 | SQL Injection |
| 4th | 79 | Cross- Site Scripting (XSS) |
| 5th | 78 | OS Command Injection |
| 6th | 319 | Cleartext Transmission of Sensitive Info. |
| 7th | 352 | Cross-site Request Forgery (CSRF) |
| 8th | 362 | Race Condition |
| 9th | 209 | Error Message Information Leak |
| 10th | 119 | Failure Memory Buffer Bounds |

TABLE 5. CWE VIEW: CWE/SANS Weaknesses in the 2010 top 25 Most Dangerous programming errors (4 categories).

| Rank | Category |
|---|---|
| - | Weaknesses On the Cusp |
| - | Porous Defenses |
| - | Risky Resource Management |
| - | Insecure Interaction Between Components |

TABLE 6. CWE VIEW: CWE/SANS Weaknesses in the 2011 top 25 Most Dangerous software errors (4 categories).

| Rank | Category |
|---|---|
| - | Weaknesses On the Cusp |
| - | Porous Defenses |
| - | Risky Resource Management |
| - | Insecure Interaction Between Components |

TABLE 7. CWS VIEW: Weaknesses in OWASP Top 10 (2013).

| Rank | Category | Title |
|---|---|---|
| 1st | A1 | Injection |
| 2nd | A2 | Broken Authent. and Session Management. |
| 3rd | A3 | Cross-Site Scripting(XSS) |
| 4th | A4 | Insecure Direct Object References |
| 5th | A5 | Security Misconfiguration |
| 6th | A6 | Sensitive Data Exposure |
| 7th | A7 | Missing Function Level Access Control |
| 8th | A8 | Cross Site Request Forgery (CSRF) |
| 9th | A9 | Using Component with Known Vul. |
| 10th | A10 | Unvalidated Redirects and Forwards |

TABLE 8. CWE VIEW: Weaknesses in OWASP Top 10 (2017).

| Rank | Category | Title |
|---|---|---|
| 1st | A1 | Injection |
| 2nd | A2 | Broken Authent. and Session Management. |
| 3rd | A3 | Sensitive Data Exposure |
| 4th | A4 | XML External Entities (XXE) |
| 5th | A5 | Broken Access Control |
| 6th | A6 | Security Misconfiguration |
| 7th | A7 | Cross-Site Scripting (XSS) |
| 8th | A8 | Insecure Deserialization |
| 9th | A9 | Using Component with Known Vulnerability. |
| 10th | A10 | Insufficient Logging and Monitoring |

## 4. MODELS DESCRIPTION

In this section, we describe the models (networks) generated using the datasets mentioned in the previous section. We generated 7 network models each of which represents the top vulnerabilities for a particular year that was published by CWE/SANS and OWASP security organizations. In complex networks, a network can be represented as nodes and relations among them. Each of the 7 datasets is used to generate a network of which the nodes are the vulnerabilities, and the edges represent the relations among these vulnerabilities. According to the description of CWE/SANS, each vulnerability has one or more related vulnerabilities that affect or can be affected by each other. Moreover, these classes define the relation of all the pairs of vulnerabilities and can be classified as follows:

– **ChildOF,** vulnerability is a child of another one.

– **ParentOf,** vulnerability is the parent of another one.

– **MemberOf,** vulnerability is in the same group of another.

– **PeerOf,** a vulnerability is a peer (or similar) of another.

According to the above classes, the relations among nodes are created. In other words, an edge is created between two nodes *if-and-only-if* one of the aforementioned kinds of relations holds. The generated

networks reflect our datasets. Each network model has its own characteristics and has a number of nodes that represent the top classified vulnerabilities of a particular year, and the number of edges represents the total number of relations among vulnerabilities. We also include the network level and nodes level measurements aiming at benchmarking our models with each other. The 7 network models are named according to the year of each list (M-2004, M-2007, M-2009, M-2010, M-2011, M-2013, and M-2017). The steps of performing this work can be summarized by the following:

**Step 1:** Preparing the datasets and defining the nodes and the edges for each dataset (vulnerabilities and their relations to others).

**Step 2:** Visualizing network models for the datasets in step 1.

**Step 3:** Calculating network measurements and the other required computations for each network model in step 2.

**Step 4:** Analyzing each network model based on the results of step 3.

**Step 5:** Concluding the obtained results in step 4.

## 5. CWE/SANS AND OWASP NETWORK MODELS

According to the description of the previous section, we generated 7 network models and summarized them in Table 9. It presents the characteristics of each network in terms of the number of nodes and edges created, *Avg(D)*, $N_D$, $D_S$, *c*, and *l*. Each of these measurements reflects a fact on a particular network. In Section 1, we presented a general description of the measurements used in this work. In our work's point of view these measurements reflect facts as follows:

- *Avg(D)*: each vulnerability is connected and has relations to other vulnerabilities, the average number of connections for all network vulnerabilities is described by this measurement.
- $N_D$: the longest path of all the shortest paths from a vulnerability to another one in the network.
- $D_S$: the number of relations among vulnerabilities. In other words, how many network vulnerabilities are related to each other.
- *c*: the tendency of network vulnerabilities to cluster together and how complete network vulnerabilities connected to each other.
- *l*: the average number of paths from a vulnerability to another one along with the shortest paths for all vulnerability's pairs.
- $C_B$: for a particular vulnerability, it represents the number of all the shortest paths from all vulnerabilities to other network vulnerabilities passing through that vulnerability. This measurement is an indicator of how influential (dangerous) a particular vulnerability in a network. The highest value of $C_B$ reflects the high risk-level of a vulnerability. Therefore, it is very important to take this measurement into considerations when ranking vulnerabilities.
- $C_C$: how close a vulnerability to other network vulnerabilities.
- $C_D$: the number of connections that a vulnerability has to other network vulnerabilities.

According to Table 9, we find the highest value of *Avg(D)* is in M-2004 meaning that most vulnerabilities under this model are vulnerable to each other. The $N_D$ and *l* of M-2010 are the highest among all the other models. This reflects the fact that most of the vulnerabilities under this model are relatively far from each other, which means these vulnerabilities are less vulnerable to each other. In M-2011 the value of *c* is the highest among the models. This means the vulnerabilities under the M-2011 model tend to cluster together, which is riskier. It should be mentioned that our models vary in terms of the number of nodes and edges. The peak of risk in the models was in 2004 in terms of *Avg*(*D*), the later years witnessed some improvements in security practices and the risk was mitigated until the year of 2011, which gained a minimum risk. However, the risk level has increased again in the years of 2013 and 2017. This is a negative indicator of the future risk of security vulnerabilities. Moreover, this pattern is observed when it comes to the other network measurements in the mentioned table.

Now, we describe and analyze each of our models separately.

TABLE 9. Characteristics of our 7 network models.

| Model | N | E | Avg(D) | $N_D$ | $D_S$ | c | *l* |
|---|---|---|---|---|---|---|---|
| M-2004 | 358 | 587 | 3.27 | 14 | 0.009 | 0.06 | 5.55 |
| M-2007 | 141 | 159 | 2.25 | 12 | 0.016 | 0.03 | 5.68 |
| M-2009 | 191 | 206 | 2.15 | 12 | 0.011 | 0.11 | 5.36 |
| M-2010 | 186 | 210 | 2.25 | 15 | 0.012 | 0.01 | 7.23 |
| M-2011 | 65 | 61 | 1.87 | 4 | 0.029 | 0.18 | 1.85 |
| M-2013 | 131 | 177 | 2.70 | 14 | 0.021 | 0.04 | 6.08 |
| M-2017 | 150 | 166 | 2.21 | 9 | 0.015 | 0.11 | 4.14 |

### A- M-2004 Model

This network model includes the OWASP top 10 vulnerabilities that are grouped into 10 categories (A1, .., A10). The order reflects the severity of risk (A1 is the

most risky). Fig. 1 depicts the M-2004 network model for each category of vulnerabilities and their related ones. The visualization shows that all the categories have relations with vulnerabilities that were not categorized in 2004. This tells us that there are many vulnerabilities not given enough attention to be considered as risky as the categorized ones since they can be a cause of other vulnerabilities. Moreover, we calculated network measurements and re-ranked the vulnerabilities according to $C_B$. As mentioned, this measurement expresses the influence of a vulnerability in its community. We also included two other measurements $C_C$ and $C_D$. Table 10 shows the proposed ranks for the vulnerabilities according to their influence on network structure. The proposed rank reveals 3 new vulnerabilities that are not under the categories of the original rank (see Table 1).

These vulnerabilities are CWE-693 *Protection Mechanisms Against Attacks*, and (CWE-668 and CWE-664) are *resource-related* vulnerabilities. These ones are not given enough attention in CWSS. Although its low degree of 11, CWE-693 has the highest value of closeness of (0.292). However, the best-connected vulnerability in our proposed rank is still the same in the original list (CWE-20). The values of $C_C$ for the other vulnerabilities seem to be close to each other, which means the vulnerabilities under the proposed list are close to each other. Based on the Girvan-Newman algorithm [15] for community detection, there are 24 communities of vulnerabilities with a modularity level of 0.7899 (strong connected communities). This is very interesting since our proposed approach includes strong communities and categories much more than what has been defined in CWSS (10 main c

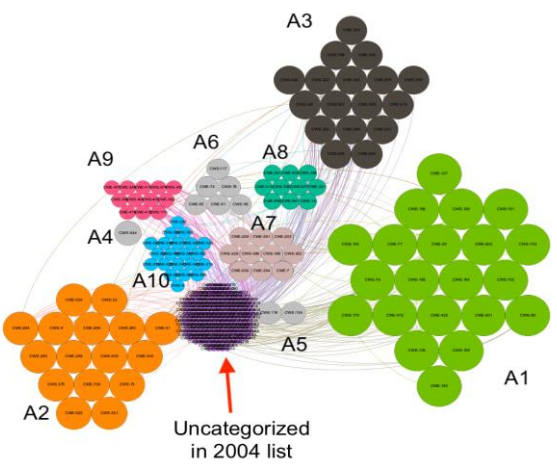


Figure 1. Visualization of M-2004 Model. Node's color reflects the category and the size reflect CWSS rank (big size of a node denotes a high rank in CWSS).

TABLE 10. M-2004 Top 10 Best Connected Vulnerabilities According to the Values of Betweenness Centrality.

| Rank/ ID | Name | Cat | $C_B$ | Cc | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 20 | Improper Input Validation | A1 | 29337 | 0.29 | 38 |
| 2nd/ 693 | Protection Mechanism Failure | - | 22291 | 0.29 | 11 |
| 3rd/ 287 | Improper Authentication | A3 | 15303 | 0.24 | 43 |
| 4th/ 284 | Improper Access Control | A2 | 14766 | 0.268 | 12 |
| 5th/ 74 | Injection | A6 | 8719 | 0.24 | 19 |
| 6th/ 668 | Exposure of Resource | - | 7459 | 0.23 | 4 |
| 7th/ 119 | Buffer Overflow | A5 | 6172 | 0.23 | 26 |
| 8th/ 404 | Improper Resource Shutdown | A9 | 6094 | 0.20 | 23 |
| 9th/ 22 | Path Traversal | A2 | 5755 | 0.24 | 9 |
| 10th/ 664 | Improper Control of a Res. | - | 5466 | 0.22 | 9 |

When we go further beyond the proposed top 10 vulnerabilities and take the ranks of 11th to 15th, we see some vulnerabilities of the original list appear using our proposed approach such as A10/CWE-552 (*Insecure Configuration Management*), A2/CWE-41 (*Broken Access Control*), A3/CWE-522 and A3/CWE-345 (*Broken Authentication and Session Management*), and A9 (*Denial of Service*). The interesting thing is that the value of betweenness centrality is significantly fallen after the mentioned vulnerabilities. This means our proposed approach does not discard the vulnerabilities in the original list; instead, it prioritizes the risk level. Therefore, our proposed approach can be integrated with the CWSS approach for providing more dimensions (more accurate) when evaluating the risk level of vulnerabilities.

### B- M-2007 Model

The network model M-2007 contains the OWASP top 10 vulnerabilities in 2007. Fig. 2 shows the visualization of the network of OWASP top 10 vulnerabilities. The figure shows the same behavior that was observed in M-2004, all the categorized depend on uncategorized vulnerabilities. It can also be seen that the categories are not strongly connected to each other. Instead, they are connected to the uncategorized vulnerabilities.

Moreover, we calculated network measurements and re-ranked vulnerabilities as presented in Table 11. Re-ranking the vulnerabilities was also based on the betweenness centrality. We compared our new ranks with the original ranks mentioned in Table 2, it can be observed that some vulnerabilities not in the top 10 categories shown in the proposed rank. These ones are CWE-668 and CWE-184 vulnerabilities. CWE-668 is *Exposure of Resources to Wrong Sphere*, and CWE-184 is *Incomplete Blacklist*. We also noticed that some of the

top-ranked vulnerabilities took the lowest ranks when using the proposed approach such as A1 and A4, while category 7 took the rank of first. The values of $C_C$ 250 reflect the same behavior as in the previous model. The number of communities detected using [15] is 12 with a modularity level of 0.788 that reflects the strength of these communities. The number of communities (categories) is increased by 2 compared to CWSS list.

In addition to the proposed top 10 ranks, we show the ranks of 11th to 15th. This range also shows uncategorized and categorized vulnerabilities (CWE-73, CWE-862, A3/CWE-434, A6/CWE-209 (*Information Leakage and Improper Error Handling*), and A10/CWE-285 (*Failure to Restrict URL Access*). The existence of these vulnerabilities in the proposed rank reflects the phenomenon mentioned in the previous model in terms of the categorized/uncategorized vulnerabilities as well as the behavior of the values of betweenness centrality.

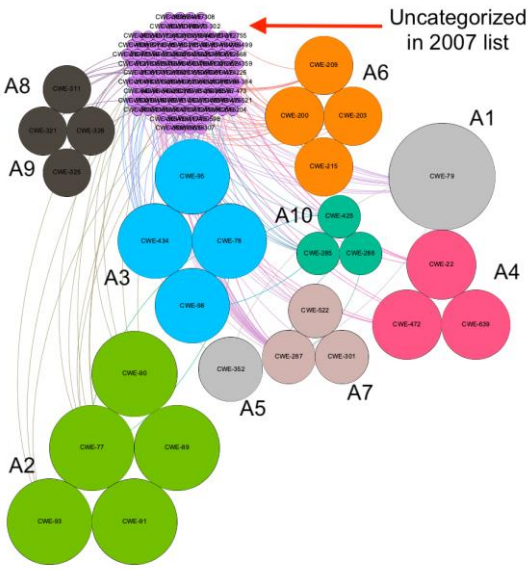


Figure 2. Visualization of M-2007 Model.

TABLE 11. M-2007 Top 10 Best Connected Vulnerabilities.

| Rank/ ID | Name | Cat | $C_B$ | Cc | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 287 | Improper Authentication | A7 | 4871 | 0.25 | 31 |
| 2nd/ 98 | Improper Control of Filename | A3 | 4087 | 0.25 | 11 |
| 3rd/ 425 | Direct Request (Forced Browsing) | A10 | 3705 | 0.24 | 8 |
| 4th/ 200 | Information Exposure | A6 | 3293 | 0.21 | 21 |
| 5th/ 668 | Exposure of Resources | - | 3292 | 0.24 | 3 |
| 6th/ 184 | Incomplete Blacklist | - | 2923 | 0.22 | 4 |
| 7th/ 288 | Authentication Bypass | A10 | 2893 | 0.25 | 6 |
| 8th/ 522 | Insufficiently Protected Credential | A7 | 2397 | 0.23 | 9 |
| 9th/ 79 | Cross-site Scripting (XSS) | A1 | 2243 | 0.19 | 14 |
| 10th/ 22 | Path Traversal | A4 | 1794 | 0.24 | 5 |

**C- M-2009 Model**

This model incorporates a combination of Microsoft SDL and CWE/SANS top vulnerabilities and is grouped into 10 categories (A1, .., A10). Fig. 3 shows the visualization of M-2009. The visualization shows similar behavior to what we have observed in the previous models. Many uncategorized vulnerabilities show an important role in the model and were not given the required attention by CWSS. Table 12 shows the proposed ranks of the vulnerabilities. The results show that 3 vulnerabilities in the proposed rank are not from the categories of the original list. These vulnerabilities are; CWE-74 *Injection*, CWE-789 *Uncontrolled Memory Allocation*, and CWE-693 *Protection Mechanism Failure*. The newly introduced vulnerabilities showed high values of $C_C$ because they are highly vulnerable to other vulnerabilities. CWE-20 has the same behavior shown in M-2004 and still in the rank of first. The community detection of Girvan detected 15 communities of vulnerabilities that are strongly connected (modularity level of 0.821). This finding introduced another 5 communities in the proposed list compared to CWSS.

Moreover, other vulnerabilities appear in the proposed list after the rank of 10th. Such that, the ranks of 11th and 12th show CWE-285 and CWE-602 (*Porous Defenses*) that are both in the original list. Also, the proposed ranks of 13th and 14th (CWE-209 (*Insecure Interaction between Components*) and CWE-665 (Risky Resource management)) are included in the original list. It should be mentioned that the 2009 original list is a combination of two lists (Table 3 and 4).

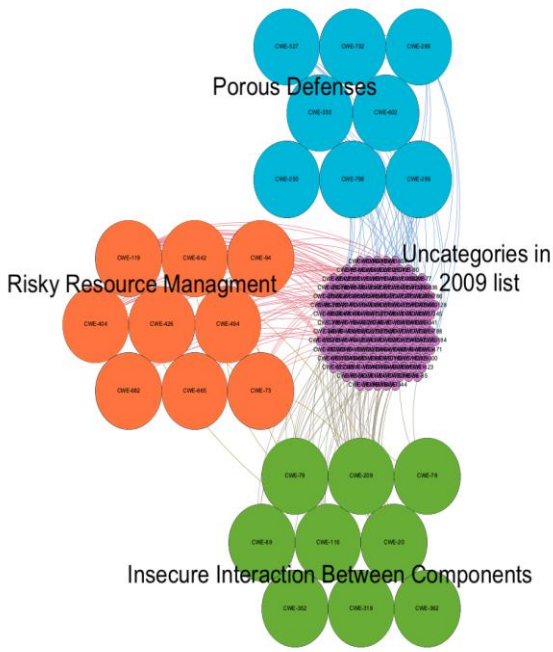


Figure 3. Visualization of M-2009 Model.

TABLE 12. M-2009 Top 10 Best Connected Vulnerabilities.

| Rank / ID | Name | Cat | $C_B$ | Cc | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 20 | Improper Input Validation | - | 9435 | 0.33 | 33 |
| 2nd/ 119 | Memory Overflow | - | 4332 | 0.27 | 25 |
| 3rd/ 74 | Injection | - | 3808 | 0.28 | 5 |
| 4th/ 642 | External Control of Cri. Data | - | 2757 | 0.23 | 8 |
| 5th/ 73 | External Control of File Name | - | 2608 | 0.27 | 10 |
| 6th/ 789 | Uncontrolled Memory Allocation | - | 2584 | 0.26 | 2 |
| 7th/ 259 | Use of Hard- Coded Password | - | 2483 | 0.22 | 5 |
| 8th/ 79 | Cross-site Scripting | - | 2315 | 0.23 | 15 |
| 9th/ 798 | Use of Hard-coded Credentials | - | 1999 | 0.18 | 6 |
| 10th/ 693 | Protection Mechanism Failure | - | 1997 | 0.26 | 3 |

## D- M-2010 Model

M-2010 model contains the CWE/SANS top 10 vulnerabilities and also grouped into 10 categories (A1, .., A10). The visualization of this model is shown in Fig. 4. It reflects similar behavior to the previous models with weak relations among the category of CWSS. The proposed ranks are shown in Table 13. The 3 new introduced vulnerabilities to the proposed list comparing to the original categories are; CWE-20 *Improper Input Validation*, CWE-668 *Exposure of Resources to Wrong Sphere*, and CWE-73 *External Control of File Name or Path*. The CWE-20 that was not classified in the CWSS of 2010, gained the highest closeness level, which reflects its severe risk. The community detection algorithm detected 15 strong communities with modularity of 0.810 in the proposed rank.

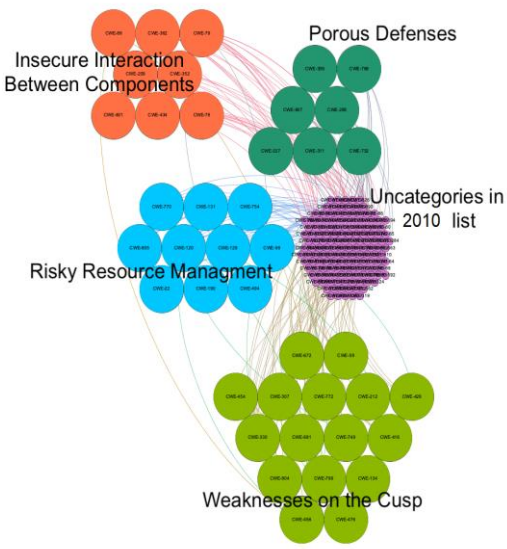


Figure 4. Visualization of M-2010 Model.

TABLE 13. M-2010 Top 10 Best Connected Vulnerabilities.

| Rank /ID | Name | Cat | $C_B$ | Cc | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 22 | Path Traversal | - | 5115 | 0.21 | 7 |
| 2nd/ 20 | Improper Input Validation | - | 4250 | 0.29 | 6 |
| 3rd/ 120 | Buffer Overflow | - | 4055 | 0.20 | 11 |
| 4th/ 732 | Incorrect Permission | - | 3910 | 0.17 | 10 |
| 5th/ 668 | Exposure of Resource | - | 3743 | 0.19 | 2 |
| 6th/ 98 | PHP Remote File Inclusion | - | 3610 | 0.19 | 11 |
| 7th/ 416 | Use After Free | - | 3533 | 0.19 | 6 |
| 8th/ 285 | Improper Autho. | - | 3069 | 0.15 | 8 |
| 9th/ 456 | Missing Init. of Var. | - | 3000 | 0.19 | 9 |
| 10th/73 | External Control of File Name | - | 2747 | 0.19 | 4 |

Furthermore, two vulnerabilities in the 2010 original list are shown in the proposed ranks of 11th and 12th for CWE-804 (*Weaknesses on the Cusp*) and CWE-434 (*Insecure Interaction Between Components*) respectively. We also observe the same behavior of what has been mentioned in 2004 model in terms of betweenness centrality values.

## E- M-2011 Model

The network model M-2011 includes the CWE/SANS top 25 vulnerabilities and grouped into 4 categories. The visualization of M-2011 shows different behavior. As presented in Table 9, M-2011 reflects better behavior compared to the other models in terms of the average degree of vulnerabilities and average path length among the vulnerabilities. We believe that CWSS list of 2011 was well-issued because the visualization showed a few uncategorized vulnerabilities in comparison to the other models (see Fig. 5). We believe this behavior reflects the collaboration with Microsoft SDL and the experience they have in this field. Table 14 lists the proposed rank of the 2011 vulnerabilities. All the vulnerabilities under each category in Table 6 appear in the proposed list except for CWE-346 *Origin Validation Error*, which takes the rank of fifth. There is no significant change comparing to the original rank. However, Girvan algorithm detected 17 communities with a modularity level of 0.856, which is interested since this model contains only 4 categories only. It should be mentioned that this model has the highest modularity level of communities compared to the other models due to the same aforementioned reason.

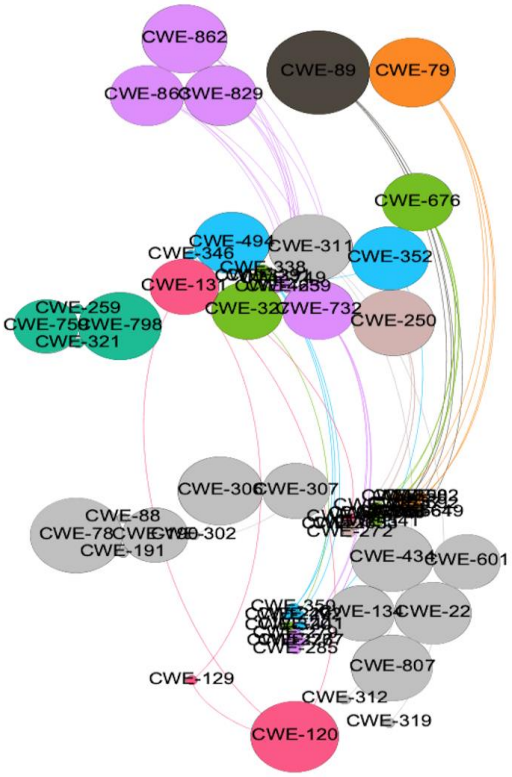


Figure 5. Visualization of M-2011 Model.

Moreover, the categorized vulnerabilities in the original list of 2011 such as CWE-352 (*Cross-Site Request Forgery (CSRF)*), CWE-329 (*Not Using a Random IV with CBC Mode*), CWE-331 (*Insufficient Entropy*), CWE-338 (*Use of Cryptographically Weak PRNG*), CWE-250 (*Execution with Unnecessary Privileges*), and CWE-131 (*Incorrect Calculation of Buffer Size*) come after the top 10 proposed ranks (11th to 16th).

TABLE 14. M-2011 Top 10 Best Connected vulnerabilities.

| Rank/ ID | Name | Cat | $C_B$ | Cc | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 732 | Incorrect Permission Assignment | - | 30 | 0.76 | 7 |
| 2nd/ 676 | Potentially Dangerous Function | - | 27 | 0.75 | 7 |
| 3rd/ 494 | Download of Code Without Check | - | 12 | 0.66 | 4 |
| 4th/ 327 | Broken/Risky Crypt. Algorithm | - | 9 | 0.5 | 4 |
| 5th/ 346 | Origin Validation Error | - | 8 | 0.6 | 2 |
| 6th/ 89 | SQL Injection | - | 6 | 1 | 4 |
| 7th/ 79 | Cross-Site Scripting | - | 6 | 1 | 4 |
| 8th/ 862 | Missing Authorization | - | 6 | 0.62 | 4 |
| 9th/ 863 | Incorrect Authorization | - | 6 | 0.62 | 4 |
| 10th/ 829 | Inclusion from Untrusted Sphere | - | 6 | 0.62 | 4 |

## F- M-2013 Model

The network model M-2013 includes the OWASP top 10 vulnerabilities and grouped into 10 categories (A1, .., A10). The visualization of the 2013 network model is shown in Fig. 6. This figure depicts clear clusters for category A2, while the others scattered in different positions in the network. The proposed rank of M-2013 is shown in Table 15. This table contains interesting results, it shows 4 vulnerabilities that are not in the categories of the original list. These vulnerabilities are; CWE-668 *Exposure of Resource to Wrong Sphere*, CWE-74 *Improper Neutralization of Special Elements in Output Used by a Downstream Component*, CWE-327 *Use of A Broken or Risky Cryptographic Algorithm*, and CWE-301 *Reflection Attack in an Authentication Protocol*. These vulnerabilities are not mentioned in any of the categories of the original 2013 list. Furthermore, we noticed that the new proposed ranks contain only the vulnerabilities that belong to the categories A2, A3, and A4 with the absence of the other categories, this means the newly introduced vulnerabilities are more vulnerable. Furthermore, M-2013 contains 18 communities and 0.809 of modularity. This is also a large number of communities compared to the other models.

The category A7 with the vulnerability of CWE-285 (*Missing Function Level Access Control*) represents the threshold before the betweenness centrality significantly falls to low values. This means most of the categorized vulnerabilities show a low level of risk in the proposed rank.

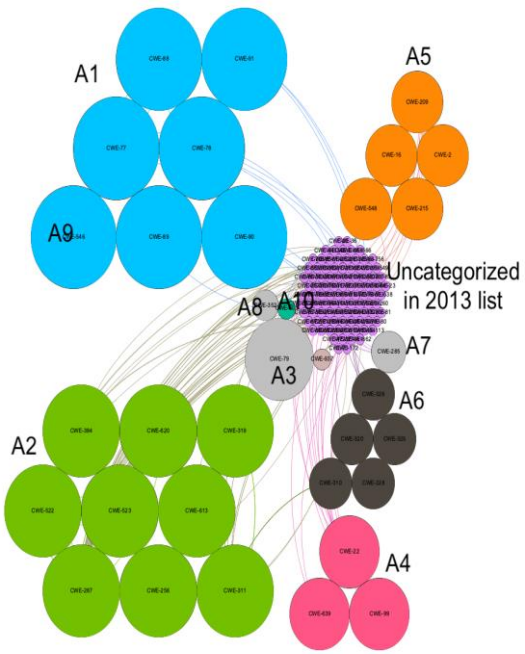


Figure 6. Visualization of M-2013 Model.

TABLE 15. M-2013 Top 10 Best Connected Vulnerabilities.

| Rank/ ID | Name | Cat | $C_B$ | $C_c$ | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 287 | Improper Authentication | *A2* | 4130 | 0.26 | 59 |
| 2nd/ 522 | Insufficiently Protected Credential. | *A2* | 2860 | 0.25 | 12 |
| 3rd/ 22 | Path Traversal | *A4* | 2601 | 0.21 | 7 |
| 4th/ 668 | Exposure of Resource | - | 2448 | 0.23 | 2 |
| 5th/ 99 | Resource Injection | *A4* | 2100 | 0.18 | 6 |
| 6th/ 74 | Injection | - | 1908 | 0.16 | 4 |
| 7th/ 327 | Broken/Risky Crypt. Algorithm | - | 1497 | 0.19 | 10 |
| 8th/ 301 | Reflection Attack an Auth. Prot. | - | 1495 | 0.22 | 3 |
| 9th/ 79 | Cross-Site Scripting | *A3* | 1255 | 0.14 | 14 |
| 10th/ 311 | Missing Encryption. of Sensitive Data | *A2* | 1141 | 0.17 | 16 |

### G- M-2017 Model

This model contains the OWASP 2017 top 10 vulnerabilities, which is the one before the last issued list. Table 16 shows the proposed ranks. It can be seen, 3 vulnerabilities appeared in the proposed rank and do not belong to the 10 categories of the 2017 list. These vulnerabilities are CWE-693 *Protection Mechanism Failure*, CWE-285 *Improper Authorization*, and CWE-668 *Exposure of Resource*. The proposed M-2017 contains 19 communities with modularity of 0.826, which is double the number of categories in the CWSS.

### H- M-2019 Model

This is the most recent list issued by CWE in 2019. It contains the top 25 most dangerous vulnerabilities according to the CWSS system. This list is issued with the support of the National Vulnerability Database (NVD). The scores in this list were based on a formula that combines the frequency of a vulnerability with the projected severity of its exploitation. Table 17 presents the recent rank of vulnerabilities.

TABLE 16. M-2017 Top 10 Best Connected Vulnerabilities.

| Rank / ID | Name | Cat | $C_B$ | $C_c$ | $C_D$ |
|---|---|---|---|---|---|
| 1st/ 287 | Improper Authentication | A2 | 2852 | 0.39 | 34 |
| 2nd/ 284 | Improper Access Control | A5 | 1802 | 0.35 | 12 |
| 3rd/ 522 | Insufficient Protected Creden | A2 | 1570 | 0.33 | 11 |
| 4th/ 693 | Protection Mechanism Failure | - | 1208 | 0.31 | 5 |
| 5th/ 327 | Broken/Risky Crypt. Algo. | A3 | 720 | 0.3 | 8 |
| 6th/ 285 | Improper Authorization | - | 702 | 0.28 | 8 |
| 7th/ 312 | Cleartext Storage of Sensitive Inf. | A3 | 654 | 0.25 | 10 |
| 8th/ 668 | Exposure of Res. | - | 608 | 0.26 | 3 |
| 9th/ 295 | Improper Certificate Validation | A3 | 537 | 0.25 | 7 |
| 10th/ 523 | UnprotectedTransport-of-Credential | A2 | 477 | 0.27 | 3 |

TABLE 17. CWE VIEW: Weaknesses in CWE top 25 vulnerabilities (2019).

| Rank | ID | Name |
|---|---|---|
| 1st | 119 | Improper Restriction of Operations within the Bounds of a Memory Buffer |
| 2nd | 79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') |
| 3rd | 20 | Improper Input Validation |
| 4th | 200 | Information Exposure |
| 5th | 125 | Out-of-bounds Read |
| 6th | 89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') |
| 7th | 416 | Use After Free |
| 8th | 190 | Integer Overflow or Wraparound |
| 9th | 352 | Cross-Site Request Forgery (CSRF) |
| 10th | 22 | Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') |
| 11th | 78 | Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') |
| 12th | 787 | Out-of-bounds Write |
| 13th | 287 | Improper Authentication |
| 14th | 476 | NULL Pointer Dereference |
| 15th | 732 | Incorrect Permission Assignment for Critical Resource |
| 16th | 434 | Unrestricted Upload of File with Dangerous Type |
| 17th | 611 | Improper Restriction of XML External Entity Reference |
| 18th | 94 | Improper Control of Generation of Code ('Code Injection') |
| 19th | 798 | Use of Hard-coded Credentials |
| 20th | 400 | Uncontrolled Resource Consumption |
| 21st | 772 | Missing Release of Resource after Effective Lifetime |
| 22nd | 426 | Untrusted Search Path |
| 23rd | 502 | Deserialization of Untrusted Data |
| 24th | 269 | Improper Privilege Management |
| 25th | 295 | Improper Certificate Validation |

Compared to the other models, M-2019 did not reflect a significant difference. This is because CWSS used almost the same approach with a little difference in evaluating the frequency and the severity of vulnerabilities. As mentioned,

## 6. DISCUSSION

According to the obtained results, we see that it is important to include more dimensions of view when assessing the risk level of vulnerabilities. Thus, integrating the proposed approach with the current scoring system can be considered as a powerful tool for software security architects. This section discusses the obtained results as follows:

### A- Network-level evaluation:

- The visualization of all the models (except M-2011) showed a common feature, which is the weak tendency of the vulnerabilities to cluster under their categories. Instead, they tend to cluster with uncategorized vulnerabilities. We see it is needed to investigate the categories and their vulnerabilities issued by CWE/SANS and OWASP. Also, it is important to consider the relations among vulnerabilities when ranking them because

vulnerabilities might become a side-effect/cause of the others. Therefore, we have witnessed introducing uncategorized vulnerabilities in the proposed ranks for all the models generated in this work.

- M-2004 has the highest *Avg(D)* of 3.27, which means there exists a high correlation among vulnerabilities in the model. This means the impact of a vulnerability is high to all the vulnerabilities in this model.
- M-2011 is the most accurate model in this work based on the obtained results. The list of 2011 is issued with the support of Microsoft SDl. Therefore, issuing such ranks should be always performed with the support of the big and well-known companies aiming at providing better practices to software developers.
- The most recent CWE list did not show a significant difference compared to the other models.

**B- Node level evaluation:**

- The resource-related vulnerability (CWE-668) can be considered as a special case and should be investigated by the security developers since it appeared in M-2004, M-2007, M-2010, M-2013, and M-2017.
- The vulnerabilities CWE-693, CWE-184, CWE-20, and CWE-346 are *validation-related* vulnerabilities and appeared in the M-2004, M-2007, M-2010, and M-2011. This means the issue of input validation and its related vulnerabilities should be taken into high considerations during the software security design phase.
- The newly introduced vulnerabilities in the proposed ranks in all 7 models are more likely to be grouped in two communities of vulnerabilities; resource-related and validation-related. Table 18 shows all the newly introduced vulnerabilities in the proposed ranks comparing to the categories of CWSS.
- It was difficult to establish categories for the *uncategorized vulnerabilities* that were appeared in the proposed ranks for all the lists. Therefore, developers can treat every single one as it is taking into consideration its relations to others.

TABLE 18. The vulnerabilities are not given enough attention either in CWSS scoring system.

| Vul. ID | Title |
|---|---|
| CWE-668 | Exposure of Resource to Wrong Sphere |
| CWE-664 | Improper Control of a Resource Through its Lifetime |
| CWE-693 | Protection Mechanism Failure |
| CWE-20 | Improper Input Validation |
| CWE-73 | External Control of File Name or Path |
| CWE-74 | Improper Neutralization of Special Elements in Output Used by a Downstream Comp. (Injection). |
| CWE-301 | Reflection Attack in an Authentication Prot. |
| CWE-327 | Use of a Broken or Risky Cryptographic Algorithm |
| CWE-346 | Origin Validation Error |
| CWE-184 | Incomplete Blacklist |
| CWE-789 | Uncontrolled memory Allocation |
| CWE-285 | Improper Authorization |

## 7. CONCLUSIONS AND FUTURE WORKS

This work formed the top security vulnerabilities for the years of 2004, 2007, 2009, 2010, 2011, 2013, and 2017 issued by CWE/SANS and OWASP organizations into 7 network models. The generated networks were formed as nodes (vulnerabilities) and edges (relations) among them. Each network model includes the top classified vulnerabilities and their categories. The aforementioned organizations use a certain scoring system (CWSS) for prioritizing and ranking vulnerabilities based on their risky level reported. In this work, we proposed an approach for prioritizing the security vulnerabilities. We used network centrality measurements in re-ranking the vulnerabilities and investigate the risk level based on their relations to others. It should be mentioned that our proposed rank is considered complementary to the current scoring system. In other words, our approach can be incorporated into the current scoring system (CWSS). Our approach involved the betweenness Centrality because it reveals the importance (influence) of a particular vulnerability in its community in terms of the relations with other vulnerabilities. Practically, this measurement fits the purpose of this work as well as it measures how much a vulnerability plays as a bridge from/to other vulnerabilities. The findings showed that using centrality measurements can be considered as an effective tool in revealing the actual risk of security vulnerabilities and it is beneficial to incorporate them in the current scoring system.

As future work, we plan to generate a temporal dynamic network model with a time interval containing all the vulnerabilities issued by CWE/SANS and OWASP. Then, reveal more in-depth facts on these vulnerabilities as well as the relations among them and show how their impact changes over time.

## ACKNOWLEDGMENT

I would like to thank the Computer Science department at the University of Mosul/ Iraq for all the support in making this research achieved. I am also grateful to the BioComplex Lab. for the support.

## REFERENCES

[1] Zhou, C.V, Leckie, C., Karunasekera, S. (2010) A survey of coordinated attacks and collaborative intrusion detection, *Computers & Security* 29 (1) 405 124–140.

[2] Wilander, J., M. Kamkar, (2003) A comparison of publicly available tools for dynamic buffer overflow prevention., in: *NDSS*, **Vol. 3**, 2003, pp. 149–162.

[3] Martin, B., Brown, M., Paller, A., Kirby, D., Christey, S. (2011) CWE/SANS top most dangerous software errors, *Common Weakness Enumeration* 7515.

[4] T. OWASP, 2004-2017, The Most Critical Web Application Security Risks (2018).

[5] Martin, B., Christey, S. (2011) Common weakness scoring system (cwss), Retrieved (date) from http://cwe. mitre. org/cwss.

[6] Younan,Y. (2013) 25 years of vulnerabilities: 1988–2012, Sourcefire Vulnerability Research Team.
[7] Watts, D. J., Strogatz, S. H. (1998) Collective dynamics of small-world networks, *nature*, **393** (6684) 440-442.
[8] Mahmood, B., Menezes, R., (2013) United states congress relations according to liberal and conservative newspapers, *Network Science Workshop* (NSW), 2013 IEEE 2nd, IEEE, 2013, pp. 98–101.
[9] Holm, H., Afridi, K.K., (2015) An expert-based investigation of the common vulnerability scoring system, *Computers & Security* **53** 18–30.
[10] Younis, Z. K., & Mahmood, B. (2020, February). Towards the Impact of Security Vunnerabilities in Software Design: A Complex Network-Based Approach. In 2020 6th International Engineering Conference "Sustainable Technology and Development"(IEC) (pp. 157-162). IEEE.
[11] Sharma, R. Singh, R. (2018) An improved scoring system for software vulnerability prioritization, *Quality, IT and Business Operations*, Springer, 33–43.
[12] Frigault, M., Wang, L., Jajodia, S., Singhal, A. (2017) Measuring the overall network security by combining cvss scores based on attack graphs and Bayesian networks, *Network Security Metrics*, Springer, 1–23.
[13] Cheng, P., Wang, L., Jajodia, S., Singhal, A. (2017) Refining cvss-based network security metrics by examining the base scores, *Network Security Metrics*, Springer, 25–52.
[14] Behi, M. GhasemiGol, M., Vahdat-Nejad, H. (2018) A new approach to quantify network security by ranking of security metrics and considering their relationships, International Journal of Network Security **20** (1) 141–148.
[15] Girvan, M., Newman, M.E. (2002) Community structure in social and biological networks, *Proceedings of the national academy of sciences* **99** (12) 7821–7826.

**Basim Mahmood** He is a faculty member at the Department of Computer Science, University of Mosul, IRAQ. He is also a member of the Biocomplex Lab., UK. He obtained his Ph.D. degree in Computer Science in the field of Complex Networks from Florida Institute of Technology/ USA in 2015. His M.Sc. degree was in Computer Science in the field of Mobile Systems from the University of Mosul/ IRAQ in 2009. His current area of research deals with Complex Networks applications, Big Data Analysis, security-related analysis and Internet of Things (IoT).